DOCUMENT RESUME

ED 075 264 24 SE 016 122

AUTHOR Suppes, Patrick
TITLE Mathematical Models of Elementary Mathematics Learning and Performance. Final Report.
INSTITUTION Stanford Univ., Calif. Inst. for Mathematical Studies in Social Science.
SPONS AGENCY National Center for Educational Research and Development (DHEW/OE), Washington, D.C. Div. of Elementary and Secondary Research.
BUREAU NO BR-0-0113
PUB DATE Feb 73
GRANT OEG-9-70-0024-057
NOTE 30p.

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS Computer Assisted Instruction; *Educational Research; *Elementary School Mathematics; *Learning; Learning Theories; Mathematical Models; Mathematics Curriculum; Mathematics Education; *Mathematics Instruction; *Research Methodology

ABSTRACT

This project was concerned with the development of mathematical models of elementary mathematics learning and performance. Probabilistic finite automata and register machines with a finite number of registers were developed as models and extensively tested with data arising from the elementary-mathematics strand curriculum developed by the Institute for Mathematical Studies in the Social Sciences and delivered to students in schools in a computer-assisted-instruction mode. Probabilistic automata were defined and tested for basic addition, subtraction, and multiplication exercises. Results of detailed application of the register machine models are unpublished and will appear subsequent to the date of this final report. The results of analyses indicate that the kinds of models developed can be applied to learning and performance in elementary mathematics, and have implications for detailed pedagogical procedures of instruction in these basic skills. (Author/DT)

FILMED FROM BEST AVAILABLE COPY

ED 075264

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

Final Report

Project No. 0-0113
Grant No. OEG-9-70-0024-057

Patrick Suppes
Institute for Mathematical Studies
in the Social Sciences
Stanford University
Stanford, California 94305

MATHEMATICAL MODELS OF ELEMENTARY MATHEMATICS LEARNING AND PERFORMANCE

February 1973

U.S. DEPARTMENT OF HEALTH, EDUCATION, AND WELFARE

Office of Education

National Center for Educational Research and Development
(Basic Research Branch, Division of Research)

SE 016 122

MATHEMATICAL MODELS OF ELEMENTARY MATHEMATICS LEARNING AND PERFORMANCE

Patrick Suppes

ABSTRACT

This project was concerned with the development of mathematical models of elementary mathematics learning and performance. Probabilistic finite automata and register machines with a finite number of registers were developed as models and extensively tested with data arising from the elementary-mathematics strand curriculum the Institute for Mathematical Studies in the Social Sciences has developed over a period of more than five years. This curriculum is delivered to students in schools at teletype terminals in a computer-assisted-instruction mode by telephone lines connected to the Institute's computer at Stanford.

Probabilistic automata were defined and tested for basic addition, subtraction and multiplication exercises. An extensive report of this work is to be found in the recent book by Patrick Suppes and Mona Morningstar entitled Computer-assisted Instruction at Stanford, 1966-68: Data, Models, and Evaluation of the Arithmetic Programs.

Detailed application of the register machine models is completed, but the results are as yet unpublished and will appear subsequent to the date of this final report.

The results of these analyses indicate that the kinds of models developed can be applied to learning and performance in elementary mathematics, and have implications for detailed pedagogical procedures of instruction in these basic skills.

Final Report

Project No. 0-0113
Grant No. OEG-9-70-0024-057

Mathematical Models of Elementary Mathematics Learning and Performance

Patrick Suppes
Stanford University
Stanford, California 94305

February 1973

The research reported herein was performed pursuant to a grant with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
National Center for Educational Research and Development

TABLE OF CONTENTS

		Page
	List of Tables	1
	List of Figures	2
I.	Introduction	3
II.	Methods	6
III.	Results	19
IV.	Conclusions	24
V.	Bibliography	25

LIST OF TABLES

Table		Page
1	Pretest Exercises in Column Addition	20
2	Maximum-likelihood Estimates of Parameters for the Three-, Four-, and Five-parameter Addition Models	22

LIST OF FIGURES

Page

Figure 1. Proportion correct per problem for data and models. 23

I. INTRODUCTION

The psychological study of arithmetic skills, like most other parts of psychology, has a relatively recent history--only a few systematic studies were made before 1890. The real impetus was provided by E. L. Thorndike's analysis of the learning of arithmetic in his Educational Psychology (1913, 1914) and later in his The Psychology of Arithmetic (1922). In an attempt to account for the acquisition of arithmetic skills in terms of his three psychological laws--the law of readiness, the law of exercise and the law of effect--he tried to justify and analyze the reason for the traditional importance attached to drill and practice in arithmetic; for him the psychological purpose of drill is to strengthen the bonds between stimuli and appropriate responses. He moved on from such fundamental questions to the more practical ones of amount and distribution of practice. He emphasized the advantages of distributed practice and criticized the actual distribution of practice in textbooks of his time. Some effects of his work on the revisions of textbooks in the 1920s and later are documented in Cronbach and Suppes (1969, pp. 103-110).

In the twenties and thirties there were a large number of good empirical studies of arithmetic skills, many of which were concerned with detailed questions that had to be answered in any complete psychological theory of arithmetic. For example, Buckingham (1925) studied student preferences and aptitudes for adding up or down in column addition problems. An extensive review of this literature may be found in Suppes, Jerman and Brian (1968).

Empirical studies like those of Buckingham were not designed to develop an overall theory of arithmetic skills; nor, it is probably fair to say, was Thorndike completely sensitive to the gap that existed between his theoretical ideas and the actual algorithms students were taught to solve problems. There are many stages to work through in developing an adequate theory, and there is no one point at which one can say the theory is now complete in all respects. If, for example, the theory is adequate at some conceptual level of information processing, then it is possible to move on to additional perceptual questions. Moreover, once a perceptual theory of a certain level of abstraction is successfully developed, it is possible to go on to still more detailed perceptual questions, such as requiring the theory to include eye movements of students as well as their numerical responses. It is an important methodological precept that at no foreseeable point shall we reach a fixed and firm bottom beyond which we cannot probe for further details and a more refined theory.

As a background for the work done under this project, we briefly sketch the history of work in the Institute for Mathematical Studies in the Social Sciences at Stanford over the past six or seven years on the psychological study of arithmetical skills. Rather than attempt a general coverage, we shall concentrate on a single example--the simple one of column addition--to illustrate how we have tried continually to deepen

the theory. The methods and results cited for this simple case are extended in the bibliography of technical publications to other arithmetical skills.

It also should be mentioned that the data used for testing the models have all been collected as part of the Institute's activities in computer-assisted instruction in elementary mathematics. The part of this work that was supported by the current project was especially directed to the elementary mathematical skills of disadvantaged students. Several of the publications, especially Suppes and Morningstar (1970), refer to early phases of this work.

The first question we tried to answer was how can one predict the relative difficulty of different exercises of column addition? If, for example, we consider problems up to the size of three columns and three rows, we are confronted with approximately one billion problems. A meaningful theory must drastically reduce this large number of exercises to a small number of classes in which all members of a class are essentially the same in difficulty.

Our first approach (Suppes, Hyman & Jerman, 1967) was to identify a small number of structural features that would permit us to apply linear regression models to predict either probability of correct response or expected latency of response. Additional applications of such regression models may be found in Suppes, Jerman and Brian (1968) and Suppes and Morningstar (1972). The application of such regression models is exemplified in Section III of this report. As can be seen from the information given there, the fit of the regression model to mean student-response data on column addition exercises is not bad. Conceptually, however, there are obvious lacunae. The regression model that predicts response probabilities does not really postulate a specific process by which students apply an algorithm to solve an exercise.

The next level of theory developed is aimed precisely at offering such process models. The natural theoretical tools for providing process models of algorithmic tasks are automata, and for most of elementary arithmetic, simple finite automata are satisfactory. There is, however, one weakness in finite automata as ordinarily defined, namely, they have no place for a probabilistic theory of error, so the natural step is to move from finite deterministic automata to probabilistic automata.

An automaton becomes probabilistic by making the transition function from state to state probabilistic in character. Thus, from a given input and a given internal state there is a probability of going to any one of several different states. In general one wants to make the output function probabilistic also. This means that given an internal state and an input there is a probability distribution over the next output. (These ideas are made formally definite in Definitions 1 and 2 of Section II.) By drastically reducing the source of error to a small number of parameters, we can develop and apply manageable probabilistic automata to student-response data.

Such a probabilistic automaton model takes a definite step beyond a regression model in providing in an abstract sense an adequate information-processing model. From a psychological standpoint, on the other hand, the automaton models are unsatisfactory in that they lack any perceptual components, and therefore they do not deal directly with how the student actually processes the format of written symbols in front of him.

Our final effort as part of this project was very much directed at this point. In principle, it would be possible to continue the development of automaton models with an abstract concept of state to represent the student's perceptual processing. A weakness of this extension of the automaton models is that when the states are left in a general abstract formulation it is natural to end up designing a different automaton for each of the different tasks in elementary mathematics, and a plethora of models results. Closer examination of the algorithmic tasks of arithmetic facing the student in solving exercises indicates that the various tasks have much in common. This commonality suggests a somewhat different approach, an approach via register machines with perceptual instructions.

Register machines were first introduced by Shepherdson and Sturgis (1963) to give a natural representation of computable functions in terms that are closer to the idea of a computer accepting instructions than to a Turing machine. In the case of the representation of computable functions, a rather simple set of arithmetic instructions is sufficient. In particular, an unlimited register machine has a denumerable sequence of registers, but any given program only uses a finite number of these registers and the machine accepts six basic instructions: add one to a register, subtract one, clear a register, copy from one register to another, and two jump instructions, one conditional and one not. (This set of six instructions is not minimal, but it is convenient.) Obviously, for the perceptual processing that a student does we want a different register machine and a radically different set of instructions. In addition, it is natural to postulate only a finite fixed number of registers that the student can use.

The basic idea of this approach is to drastically simplify the perceptual situation by conceiving each exercise as being presented on a grid. The student is represented by a model that has instructions for attending to a given square on the grid; for example, in the standard algorithms of addition, subtraction and multiplication we begin in the upper right-hand corner and then have instructions to move downward through each column and from right to left across columns. Additional instructions for storing the results of an operation, for outputting the last digit of a stored numeral, etc., are needed. Some further details are given in Section II, but the discussion is not as complete as that for automaton models.

The basic idea of register machines is that the different algorithms are represented by subroutines. One subroutine may be called in another, as complex routines are built up. The procedure is familiar to most of

us. For example, in performing column multiplication we use the algorithm of addition, which in this case means calling the subroutine for addition; in long division we call the subroutines for subtraction and multiplication, as well as for addition. Each basic subroutine is represented by a program in terms of the primitive instructions. The problem from a psychological standpoint is to find instructions that provide not only a realistic description of what the student does, a description that can be fitted to data in the same way that the automaton models have been applied to data, but also a fuller account of how the student processes the exercise.

At the first stage of analyzing register machine models we can get results similar to those for the automaton models by postulating error parameters for execution of main subroutines of the routine for a given algorithm. In addition to providing an explicit analysis of perceptual processing, the register machines provide a natural device for analyzing learning. However, the detailed and technical extension of register machines to learning was not made during the course of the present project, but is work planned for the future.

In Section II we describe the theoretical and empirical methods used in the project and give a brief, but technical, description of the most important models tested. Finally, in Section III we summarize the results of the test of the models, but because of the great technical detail of the results this report provides only a summary, and we refer the reader to publications that have already appeared or are forthcoming, if he wishes to pursue the details.

II. METHODS

Although the bulk of the research conducted under this project was concerned with probabilistic automaton and register machine models, we extended our earlier work on linear regression models and begin this discussion of the models with a presentation of the linear regression model for column addition reported in Suppes and Morningstar (1972).

Linear regression models. We began with regression models that use as independent variables structural features of individual arithmetic exercises. We denote the j^{th} structural feature of exericse i in a given set of exercises by f_{ij}. The parameters estimated from the data are the values attached to each structural feature. (In previous publications we have referred to these structural features as factors, but this can lead to confusion with the concept of factor as used in factor analysis.) We denote the coefficient assigned to the j^{th} structural feature by α_j, and we emphasize that the structural features themselves, as opposed to their coefficients, are objectively identifiable by the experimenter in terms of the exercises themselves, independent of the response data.

Let p_i be the observed proportion of correct responses on exercise i for a given group of students. The natural linear regression in terms of the structural features f_{ij} and the coefficients α_j is simply

$$p_i = \sum_j \alpha_j f_{ij} + \alpha_0 .$$

Unfortunately, when the regression is put in this form, there is no guarantee that probability will be preserved as the structural features are combined to predict the observed proportion of correct responses. To guarantee conservation of probability, it is natural to make the following transformation and to define a new variable z_i.

$$(1) \qquad z_i = \log \frac{1 - p_i}{p_i} ,$$

and then to use as the regression model

$$(2) \qquad z_i = \sum_j \alpha_j f_{ij} + \alpha_0 .$$

The numerator of equation (1) contains $1 - p_i$ rather than p_i, so that the variable z_i increases monotonically rather than decreases monotonically with the magnitude of the structural features f_{ij}.

In Chapter 3 of Suppes and Morningstar (1972), the following structural features were defined for column-addition exercises.

The feature SUMR is the number of columns in the largest addend. For three-row exercises SUMR is defined as 1.5 times the number of columns, plus .5 if a column sum is 20 or more. For example,

$$\text{SUMR} \left(\begin{array}{r} a \\ \underline{+\ b} \\ c \end{array} \right) = 1$$

$$\text{SUMR} \left(\begin{array}{r} a \\ b \\ \underline{+\ c} \\ de \end{array} \right) = \begin{cases} 1.5 & \text{if } de < 20 \\ 2 & \text{if } de \geq 20 \end{cases}$$

$$\text{SUMR} \left(\underline{ab} + c = de \right) = 2 .$$

This structural feature reflects the number of columns of addition, with greater weight being given to columns in three-row exercises than in two-row exercises.

The second structural feature is CAR, which represents the number of times the sum of a column, including any numbers carried to it, exceeds nine. For example,

$$\mathrm{CAR}\left(\begin{array}{r} a \\ +\ b \\ \hline c \end{array}\right) = 0$$

$$\mathrm{CAR}\,(a + b = \underline{cd}) = 1$$

$$\mathrm{CAR}\left(\begin{array}{r} ab \\ +\ cd \\ \hline ef \end{array}\right) = \begin{cases} 0 & \text{if } b + d \leq 9 \\ 1 & \text{if } b + d > 9 \end{cases}$$

$$\mathrm{CAR}\left(\begin{array}{r} ab \\ cd \\ +\ ef \\ \hline ghi \end{array}\right) = \begin{cases} 1 & \text{if } b + d + f \leq 9,\ a + c + e > 9 \\ 2 & \text{if } b + d + f > 9,\ a + c + e \geq 9\,. \end{cases}$$

The third structural feature VF reflects the vertical format of the exercise. The vertical exercises with one-digit responses were given the value 0. Multicolumn exercises with multidigit exercises and one-column addition exercises with a response of 11 were given the value 1. One-column addition exercises with a multidigit response other than 11 were given the value 3. For example,

$$\mathrm{VF}\left(\begin{array}{r} ab \\ -\ cd \\ \hline e \end{array}\right) = 0$$

$$\mathrm{VF}\left(\begin{array}{r} abc \\ +\ def \\ \hline ghi \end{array}\right) = 1$$

$$\mathrm{VF}\left(\begin{array}{r} a \\ +\ b \\ \hline cd \end{array}\right) = 3\,.$$

This structural feature is meant to reflect the likelihood of the mistake of reversing the digits of the correct response, especially in a one-column addition exercise. In the computer-assisted instruction environment where students were responding at teletype terminals, responses to vertical exercises were typed from right to left, while responses to horizontal exercises were typed from left to right. Thus, it was possible for a student to have in mind the correct answer, but to err by typing the digits in the reverse order. It is fair to say that this structural feature is of more importance in working at a computer-based terminal than when using paper and pencil.

Three-state automaton model. The central weakness of the regression models is that they are not process models. They do not provide even a schematic analysis of the algorithmic steps the student uses to find an answer. Automaton models are process models and therefore their use represents a natural extension of the regression analysis. For the exercises in column addition we may restrict ourselves to finite automata, but as ordinarily defined they have no place for errors. However, this is easily introduced by moving from deterministic state transitions to probabilistic ones.

We begin with the definition of a finite deterministic automaton, and then generalize.

Definition 1. A structure $\mathfrak{A} = \langle A, V_I, V_O, M, Q, s_0 \rangle$ is a finite (deterministic) automaton with output if and only if

(i) A is a finite, nonempty set,

(ii) V_I and V_O are finite nonempty sets (the input and output vocabularies, respectively),

(iii) M is a function from the Cartesian product $A \times V_I$ to A (M defines the transition table),

(iv) Q is a function from the Cartesian product $A \times V_I$ to V_O (Q is the output function),

(v) s_0 is in A (s_0 is the initial state).

As an example of a finite automaton with output, that is, a finite automaton in the sense of this definition, we may characterize an automaton that will perform two-row column addition.

$$A = \{0,1\} ,$$

$$V_I = \{(m,n) : 0 \leq m, n \leq 9\} ,$$

$$V_O = \{0,1,\ldots,9\} ,$$

$$M(k,(m,n)) = \begin{cases} 0 & \text{if } m + n + k \leq 9 , \\ 1 & \text{if } m + n + k > 9 , \end{cases} \quad \text{for } k = 0,1 ,$$

$$Q(k,(m,n)) = (k + m + n) \bmod 10 ,$$

$$s_0 = 0.$$

Thus the automaton operates by adding first the ones' column, storing as internal state 0 if there is no carry, 1 if there is a carry,

outputting the sum of the ones' column modulus 10, and then moving on to the input of the two tens' column digits, etc. The initial internal state s_0 is 0, because at the beginning of the exercise there is no 'carry'.

Definition 2. A structure $\mathfrak{A} = \langle A, V_I, V_O, p, q, s_0 \rangle$ is a (finite) probabilistic automaton if and only if

(i) A is a finite, nonempty set,

(ii) V_I and V_O are finite, nonempty sets,

(iii) p is a function on $A \times V \times A$ to the interval $[0,1]$ such that for each s in A and σ in V, $p_{s,\sigma}$ is a probability density over A, i.e.,

(a) for each s' in A, $p_{s,\sigma}(s') \geq 0$,

(b) $\sum_{s' \in A} p_{s,\sigma}(s') = 1$,

(iv) q is a function on $A \times V_I \times V_O$ to $[0,1]$ such that for each s in A and σ in V, $q_{s,\sigma}$ is a probability density over V_O,

(v) s_0 is in A.

In the probabilistic generalization of the automaton for column addition, the number of possible parameters that can be introduced is uninterestingly large. Each transition $M(k,(m,n))$ may be replaced by a probabilistic transition $1 - \epsilon_{k,m,n}$ and $\epsilon_{k,m,n}$, and each output $Q(k(m,n))$, by 10 probabilities for a total of 2200 parameters.

A three-parameter automaton model structurally rather close to the regression model is easily defined. First, two parameters, ϵ and η, are introduced according to whether there is a 'carry' to the next column.

$$P(M(k,(m,n)) = 0 \mid k + m + n \leq 9) = 1 - \epsilon$$

and

$$P(M(k,(m,n)) = \overline{1} \mid k + m + n > 9) = 1 - \eta .$$

In other words, if there is no 'carry', the probability of a correct transition is $1 - \epsilon$ and if there is a 'carry' the probability of such a transition is $1 - \eta$. The third parameter, γ, is simply the probability of an output error. Conversely, the probability of a correct output is:

$$P(Q(k,(m,n)) = (k + m + n) \bmod 10) = 1 - \gamma .$$

Consider now exercise i with C_i carrys and D_i digits. If we ignore the probability of two errors leading to a correct response (e.g., a transition error followed by an output error), then the probability of a correct answer is just

$$(3)\quad P(\text{Correct Answer to Exercise } i) = (1-\gamma)^{D_i}(1-\eta)^{C_i}(1-\epsilon)^{D_i-C_i-1} .$$

As already indicated, it is important to realize that this equation is an approximation of the 'true' probability. However, to compute the exact probability it is necessary to make a definite assumption about how the probability γ of an output error is distributed among the nine possible wrong responses. A simple and intuitively appealing one-parameter model is the one that arranges the 10 digits on a circle in natural order with 9 next to 0, and then makes the probability of an error j steps to the right or left of the correct response δ^j. For example, if 5 is the correct digit, then the probability of responding 4 is δ, of 3 is δ^2, of 2 is δ^3, of 1 is δ^4, of 0 is δ^5, of 6 is δ, of 7 is δ^2, etc. Thus in terms of the original model

$$\gamma = 2(\delta + \delta^2 + \delta^3 + \delta^4) + \delta^5 .$$

Consider now the exercise

$$\begin{array}{r} 47 \\ \underline{+\ 15} \end{array} .$$

Then, where d_i = the i^{th} digit response,

$$P(d_1 = 2) = (1-\gamma) ,$$

$$P(d_2 = 6) = (1-\gamma)(1-\eta) + \eta\delta .$$

Here the additional term is $\eta\delta$, because if the state entered is 0 rather than 1 when the pair (7,5) is input, the only way of obtaining a correct answer is for 6 to be given as the sum of $0 + 4 + 1$, which has a probability δ. Thus the probability of a correct response to this exercise is $(1-\gamma)[(1-\gamma)(1-\eta) + \eta\delta]$. Hereafter we shall ignore the $\eta\delta$ (or $\epsilon\delta$) terms.

We may get a direct comparison with the linear regression model if we take the logarithm of both sides of (3) to obtain:

$$(4)\quad \log p_i = D_i \log(1-\gamma) + C_i \log(1-\eta) + (D_i - C_i - 1)\log(1-\epsilon) ,$$

and estimate $\log 1-\gamma$, $\log 1-\eta$, and $\log 1-\epsilon$ by regression with the additive constant set equal to zero. We also may use some other approach to estimation such as minimum χ^2 or maximum likelihood.

The automaton model naturally suggests a more detailed analysis of the data. Unlike the regression model, the automaton provides an immediate analysis of the digit-by-digit responses. Ignoring the $\epsilon\delta$-type terms, we can in fact find the general maximum-likelihood estimates of γ, ϵ, and η when the response data are given in this more explicit form.

Let there be n digit responses in a block of exercises. For $1 \leq i \leq n$ let x_i be the random variable that assumes the value 1 if the i^{th} response is correct and 0 otherwise. It is then easy to see that

$$P(x_i = 1) = \begin{cases} (1 - \gamma) & \text{if } i \text{ is a ones'-column digit,} \\ (1 - \gamma)(1 - \epsilon) & \text{if it is not a ones' column and there is no carry to the } i^{th} \text{ digit,} \\ (1 - \gamma)(1 - \eta) & \text{if there is a carry to the } i^{th} \text{ digit,} \end{cases}$$

granted that $\epsilon\delta$-type terms are ignored. Similarly for the same three alternatives

$$P(x_i = 0) = \begin{cases} \gamma \\ 1 - (1 - \gamma)(1 - \epsilon) \\ 1 - (1 - \gamma)(1 - \eta) \, . \end{cases}$$

So for a string of actual digit responses $x_1,\ldots,x_n$ we can write the likelihood function as:

$$(5) \quad L(x_1,\ldots,x_n) = (1-\gamma)^a\gamma^b(1-\epsilon)^c(1-\eta)^d[1-(1-\gamma)(1-\epsilon)]^e[1-(1-\gamma)(1-\eta)]^f,$$

where a = number of correct responses, b = number of incorrect responses in the ones' column, c = number of correct responses not in the ones' column when the internal state is 0, d = number of correct responses when the internal state is 1, e = number of incorrect responses not in the ones' column when the internal state is 0, and f = number of incorrect responses when the internal state is 1. (In the model statistical independence of responses is assured by the correction procedure.) It is more convenient to estimate $\gamma' = 1 - \gamma$, $\epsilon' = 1 - \epsilon$, and $\eta' = 1 - \eta$. Making this change, taking the logarithm of both sides of (5) and differentiating with respect to each of the variables, we obtain three equations that determine the maximum-likelihood estimates of γ', ϵ', and η':

$$\frac{\partial L}{\partial \gamma'} = \frac{a}{\gamma'} - \frac{b}{1 - \gamma'} - \frac{e\epsilon'}{1 - \gamma'\epsilon'} - \frac{f\eta'}{1 - \gamma'\eta'} = 0 \, ,$$

$$\frac{\partial L}{\partial \epsilon'} = \frac{c}{\epsilon'} - \frac{e\gamma'}{1 - \gamma'\epsilon'} = 0 ,$$

$$\frac{\partial L}{\partial \eta'} = \frac{d}{\eta'} - \frac{f\gamma'}{1 - \gamma'\eta'} = 0 .$$

Solving these equations, we obtain as estimates:

$$\gamma' = \frac{a - c - d}{a + b - c - d} ,$$

$$\hat{\epsilon}' = \frac{c(a + b - c - d)}{(c + e)(a - c - d)} ,$$

$$\hat{\eta}' = \frac{d(a + b - c - d)}{(d + f)(a - c - d)} .$$

Analysis of data using these estimates is discussed in the next section.

Four-parameter and five-parameter automaton models. As a result of preliminary work with the data selected to test our models, we considered two expansions of the basic automaton model. The first is a four-parameter model that generalizes the basic three-parameter model to account for some contextual effects that were observed in student responses to addition problems with consecutive carry and no-carry receiving columns. The second model generalizes the four-parameter model further to create a fifth parameter. This extension allows the model to treat addition columns that consist of only one digit differently from normal columns that consist of pairs of digits to be added.

The four-parameter model is constructed by splitting parameter ϵ, defined above, into two parameters denoted by ϵ_0 and ϵ_1. This is done to distinguish addition columns which, when correctly solved, do not receive a carry from an immediately preceding column, although that predecessor does receive a carry. Thus, the model now reflects the possible influence of the present state (carry) upon a student's tendency to generate unnecessary carrys. The reasons for this modification are made clearer when we discuss the analysis of the data. Parameters γ and η are retained as they were defined in the three-parameter model. The definition of ϵ is replaced by

$$\epsilon_0 = p[c_{n+1} = 1 \mid c_n = 0 \text{ and } x_n + y_n < 10] ,$$

$$\epsilon_1 = p[c_{n+1} = 1 \mid c_n = 1 \text{ and } x_n + y_n < 10] ,$$

where c_i is the carry (0 or 1) to column i of the problem.

The four-parameter model is also a two-state machine, but with one state transition probability now a function of the state value.

The probability that any given addition problem will be solved correctly may be written:

(6) P(All Answer Digits Correct)

$$= (1 - \gamma)^{D}(1 - \eta)^{C}(1 - \epsilon_0)^{D-C-E-1}(1 - \epsilon_1)^{E},$$

where D and C are the number of digits and carrys as defined earlier, and E is the number of columns in the problem that do not receive a carry, but which are preceded by a column that does. We continue to assume that all errors occur independently and yet do not combine to produce correct answers. This assumption applies to all the automaton models we discuss.

We shall not reproduce the details of the likelihood equation for the four-parameter model. These are given in Suppes and Morningstar (1972, p. 144).

The five-parameter model is directly obtained from the preceding models by splitting the output-error probability γ into two components. The generality of the four-parameter model and its parameters ϵ_0, ϵ_1, and η are retained, but we add:

γ_0 = the probability that an output error occurs given that either only one digit is printed in the column to be added, or a carry from the preceding column is the only source of output and no printed digits appear in the column;

γ_1 = the probability that an output error occurs given that more than one digit is printed in the column to be added, or that one printed digit and a carry from the preceding column occur.

This modification allows the model to reflect the conjecture that students find it easier to process an addition column in which all they need do to get the correct answer digit is to copy the only digit appearing in that column; this, as opposed to adding two digits and perhaps a carry before obtaining an answer.

The probability that an answer to a given problem will be correct may be written as:

(7) P(All Answer Digits Correct)

$$= (1 - \gamma_0)^{D-S}(1 - \gamma_1)^{S}(1 - \eta)^{C}(1 - \epsilon_0)^{D-C-E-1}(1 - \epsilon_1)^{E},$$

where D, C, and E are as defined earlier, and S is the number of columns in the problem which involve the sum of at least two digits, or one digit and a carry. Again, for details of estimation problems in this model, the reader is referred to Suppes and Morningstar (1972, pp. 147-148).

Register machines with perceptual instructions. To introduce greater generality and to deepen the analysis to include specific ideas about the perceptual processing of a column-addition exercise, we moved on to register machines for the reasons already described in Section I.

For column addition three registers suffice in our scheme of analysis. First there is the stimulus-supported register [SS] that holds an encoded representation of a printed symbol to which the student is perceptually attending. In the present case the alphabet of such symbols consists of the 10 digits and the underline symbol '_'. As a new symbol is attended to, previously stored symbols are lost unless transferred to a non-stimulus-supported register. The second register is the non-stimulus-supported register [NSS]. It provides long-term storage for computational results. The third register is the operations register [OP] that acts as a short-term store, both for encodings of external stimuli and for results of calculations carried out on the contents of other registers. It is also primarily non-stimulus-supported.

As already stated in the main text, we drastically simplify the perceptual situation by conceiving each exercise as being presented on a grid with at most one symbol in each square of the grid. For column addition we number the coordinates of the grid from the upper right-hand corner. Thus, in the exercise

$$\begin{array}{r} 15 \\ 24 \\ +\ 37 \\ \hline \end{array}$$

the coordinates of the digit 5 are (1,1), the coordinates of 4 are (2,1), the coordinates of 7 are (3,1), the coordinates of 1 are (1,2) and so forth, with the first coordinate being the row number and the second being the column number.

The restricted set of instructions we need for column addition are the following 10.

Attend (a,b):	Direct attention to grid position (a,b).
(±a, ±b):	Shift attention on the grid by (±a, ±b).
Readin [SS]:	Read into the stimulus-supported register the physical symbol in the grid position addressed by Attend.
Lookup [R1] + [R2]:	Look up table of basic addition facts for adding contents of register [R1] and [R2] and store the result in [R1].
Copy [R1] in [R2]:	Copy the content of register [R1] in register [R2].
Deleteright [R]:	Delete the rightmost symbol of register [R].

Jump L: Jump to line labeled L.

Jump (val) R,L: Jump to line labeled L if content of register [R] is val.

Outright [R]: Write (output) the rightmost symbol of register [R] at grid position addressed by Attend.

End: Terminate processing of current exercise.

Exit: Terminate subroutine processing and return to next line of main program.

Of the 10 instructions only Lookup does not have an elementary character. In our complete analysis it has the status of a subroutine built up from more primitive operations such as those of counting. It is, of course, more than a problem of constructing the table of basic addition facts from counting subroutines; it is also a matter of being able to add a single digit to any number stored in the non-stimulus-supported register [NSS] or [OP], as, for example, in adding many rows of digits in a given column. I omit the details of building up this subroutine.

It should also be obvious that the remaining nine instructions are not a minimal set; for example, the unconditional jump instruction is easily eliminated. We do think the nine are both elementary and psychologically intuitive for the subject matter at hand.

To illustrate in a simple way the use of subroutines, we may consider two that are useful in writing the program for column addition. The first is the vertical scan subroutine, which is needed for the following purpose. In adding rows of numbers with an uneven number of digits, we cannot simply stop when we reach a blank grid square on the left of the topmost row. We must also scan downward to see if there are digits in that column in any other row. A second aspect of this same problem is that in our model the student is perceptually processing only one grid square at a time, so that he must have a check for finding the bottom row by looking continually for an underline symbol. Otherwise he could, according to an apparently natural subroutine, proceed indefinitely far downward encountering only blanks and leaving entirely the immediate perceptual region of the formatted exercise. Here is the subroutine. In the main program it is preceded by an Attend instruction.

Vertical Scan Subroutine

```
V-scan (0-9,_)
    Rd      Readin
            Jump (0-9,_) SS, Fin
            Attend (+1,-1)
            Readin
```

```
            Jump (_) SS, Fin
            Attend (+0,+1)
            Jump Rd
Fin         Exit
```

The labels Rd and Fin of two of the lines are shown on the left.

The second subroutine is one that outputs all the digits in a register working from right to left. For example, in column addition, after the leftmost column has been added, there may still be several digits remaining to print out to the left of this column in the 'answer' row.

```
Output [R]
    Put     Outright [R]
            Deleteright [R]
            Attend (0,+1)
            Jump (Blank) R, Fin
            Jump Put
    Fin     Exit
```

Using these two subroutines the program for vertical addition is relatively straightforward and requires 26 lines. I number the lines for later reference; they are not a part of the program.

```
        Vertical Addition

1.              Attend (1,1)
2.              Readin
3.              Copy [SS] in [OP]
4.              Attend (+1,+0)
5.              Readin
6.      Opr     Lookup [OP] + [SS]
7.      Rd      Attend (+1,0)
8.              Readin
9.              Jump (0-9) SS, Opr
```

```
10.                 Jump (Blank) SS, Rd
11.                 Attend (+1,0)
12.                 Outright [OP]
13.                 Deleteright [OP]
14.                 Copy [OP] in [NSS]
15.                 Attend (1,+1)
16.                 V-scan (0-9,_)
17.                 Jump (_) SS, Fin
18.                 Jump (0-9) SS, Car
19.                 Copy [SS] in [OP]
20.                 Jump Rd
21.         Car     Copy [NSS] in [OP]
22.                 Jump Opr
23.         Fin     Jump (Blank) NSS, Out
24.                 Attend (+1,0)
25.                 Output [NSS]
26.         Out     End
```

To show how the program works, we may consider a simple one-column addition exercise. We show at the right of each line the content of each register just before the next row is attended to, i.e., after all operations have been performed.

	[SS]	[OP]	[NSS]
4	4	4	
5	5	9	
3	3	12	
8	8	20	
—	—	20	
0	0		2

This kind of analysis can be generalized to prove that the program is correct, i.e., will output the correct answer to any column-addition exercise, but this aspect of matters will not be pursued further here.

By attaching error parameters to various segments of the program, performance models are easily generated. For comparative purposes we

may define a performance model essentially identical to the two-state probabilistic automaton already introduced for column addition restricted to two rows. To lines 6-12 we attach the output error parameter γ, and to lines 13-19 we attach the 'carry' error parameter η if there is a carry, and the error parameter ϵ if there is not. Given this characterization of the error parameters the two performance models are behaviorally identical. On the other hand, it is clear that the program for the three-register machine is much more general than the two-state probabilistic automaton, since it is able to solve any vertical addition exercise. It is also obvious that other performance models can easily be defined for vertical addition by introducing error parameters attached to different segments of the program.

Students. To test the models outlined above, as already indicated, we used students participating in the elementary-mathematics CAI programs of the Institute. This project supported 10 terminals located in Brentwood School in East Palo Alto. More than 80 percent of the students attending Brentwood School are minority students. Much of the data reported in publications listed below are drawn from Brentwood School. In addition, the models described above were also applied toward the end of the grant period to data drawn from handicapped children participating in the CAI network organized by the Institute to include a number of schools for deaf students.

III. RESULTS

We summarize in this section the main results. We emphasize, however, that the reader who is interested in the detailed testing of the models is urged to go to the publications summarized at the end of this section in which results are reported. For simplicity, we do not consider the testing of the register machine models but concentrate here on the linear regression models and the probabilistic automaton models.

Linear regression models. Using the linear regression model described in Section II, the following regression was obtained for the mean response data of 63 third graders taking the pretest shown in Table 1. Complete experimental details about the students, etc., are given in Suppes and Morningstar (1972, Ch. 3).

$$p_i = .53\ \mathrm{SUMR}_i + .93\ \mathrm{CAR}_i + .31\ \mathrm{VF}_i - 4.06\ .$$

The multiple R was .74 and R^2 was .54, which reflects a reasonable fit to the data. Extensive data analysis of a similar kind for other basic skills, using similar regression models, is reported in Suppes and Morningstar (1972).

Test of automaton models for addition. Tests of the three-, four- and five-parameter addition models for third-grade addition, including

TABLE 1

Pretest Exercises in Column Addition

1)	17 + 2	8)	11 22 + 14	15)	5267 + 283
2)	6 6 + 5	9)	27 + 4	16)	46 75 + 23
3)	14 + 15	10)	8 + 32	17)	3986 + 4735
4)	6 + 13	11)	639 + 212	18)	27 46 + 88
5)	363 + 214	12)	66 + 14	19)	7657 + 1875
6)	416 + 212	13)	378 + 125	20)	69 36 + 48
7)	12 31 + 10	14)	557 + 256		

the exercises displayed in Table 1, as well as similar fourth-grade exercises, are shown in Table 2. Both sets of exercises were given both as pretests and posttests, and the corresponding parameter estimates are shown in the table.

The fit of the predictions to observed proportion correct is shown in Figure 1.

These data, reported in a 70-page chapter of Suppes and Morningstar (1972), were analyzed in conjunction with Mr. Alex Cannara. Interested readers are referred to Chapter 4 of that book.

Publications. Research performed under this grant has been published in the following articles.

P. Suppes and M. Morningstar. Four programs in computer-assisted instruction. In W. H. Holtzman (Ed.), Computer-assisted instruction, testing, and guidance. New York: Harper & Row, 1970. Pp. 233-265.

P. Suppes and M. Morningstar. Technological innovations: Computer-assisted instruction and compensatory education. In F. Korten, S. Cook & J. Lacey (Eds.), Psychology and the problems of society. Washington, D. C.: American Psychological Association, Inc., 1970. Pp. 221-236.

P. Suppes. Computer-assisted instruction at Stanford. In Man and computer. Proceedings of international conference, Bordeaux 1970. Basel: Karger, 1972. Pp. 298-330.

P. Suppes and M. Morningstar. Computer-assisted instruction at Stanford, 1966-68: Data, models, and evaluation of the arithmetic programs. New York: Academic Press, 1972, 533 pp.

P. Suppes. Facts and fantasies of education. In M. C. Wittrock (Ed.), Changing schools: Alternatives from educational research. Englewood Cliffs, N. J.: Prentice-Hall, 1973. (The technical appendix of this article reports research conducted under this grant.)

Publications in preparation. The principal investigator, Patrick Suppes, together with his collaborators in the Institute, are in the process of preparing a book that will be a sequel to the Suppes and Morningstar (1972) book, and that will report a number of detailed analyses of the models described in this report. There will be an especially long chapter on the register machine models written by Suppes and Lindsay Flannery, a graduate student in the Institute. Part of the results on register machine models of Suppes and Flannery will also be written for publication in article form, probably in the Journal of Mathematical Psychology.

TABLE 2

Maximum-likelihood Estimates of Parameters for the Three-, Four-, and Five-parameter Addition Models

	Block 306				Block 404			
	Pretest		Posttest		Pretest		Posttest	
Model parameters \ Number of students	Form A	Form B	Form B	Form A	Form A	Form B	Form B	Form A
	63	59	63	59	53	77	53	77
γ	.0430	.0364	.0216	.0400	.0290	.0252	.0154	.0128
ϵ	.0085	.0117	.0311	.0095	.0089	0	0	.0185
η	.0576	.0608	.0621	.0665	.0599	.0610	.0145	.0346
γ	.0385	.0354	.0215	.0340	.0257	.0252	.0154	.0128
ϵ_0	0	0	.0248	0	0	0	0	.0030
ϵ_1	.0928	.0773	.0591	.1130	.2470	-	-	.2212
η	.0622	.0674	.0621	.0722	.0631	.0610	.0145	.0346
γ_0	.0340	.0027	0	.0180	.0140	.0187	.0107	.0082
γ_1	.0389	.0365	.0241	.0356	.0286	.0262	.0160	.0129
ϵ_0	0	.0087	.0256	0	0	0	0	.0052
ϵ_1	.0945	.0764	.0569	.1266	.2415	-	-	.2200
η	.0616	.0629	.0601	.0674	.0605	.0610	.0139	.0336

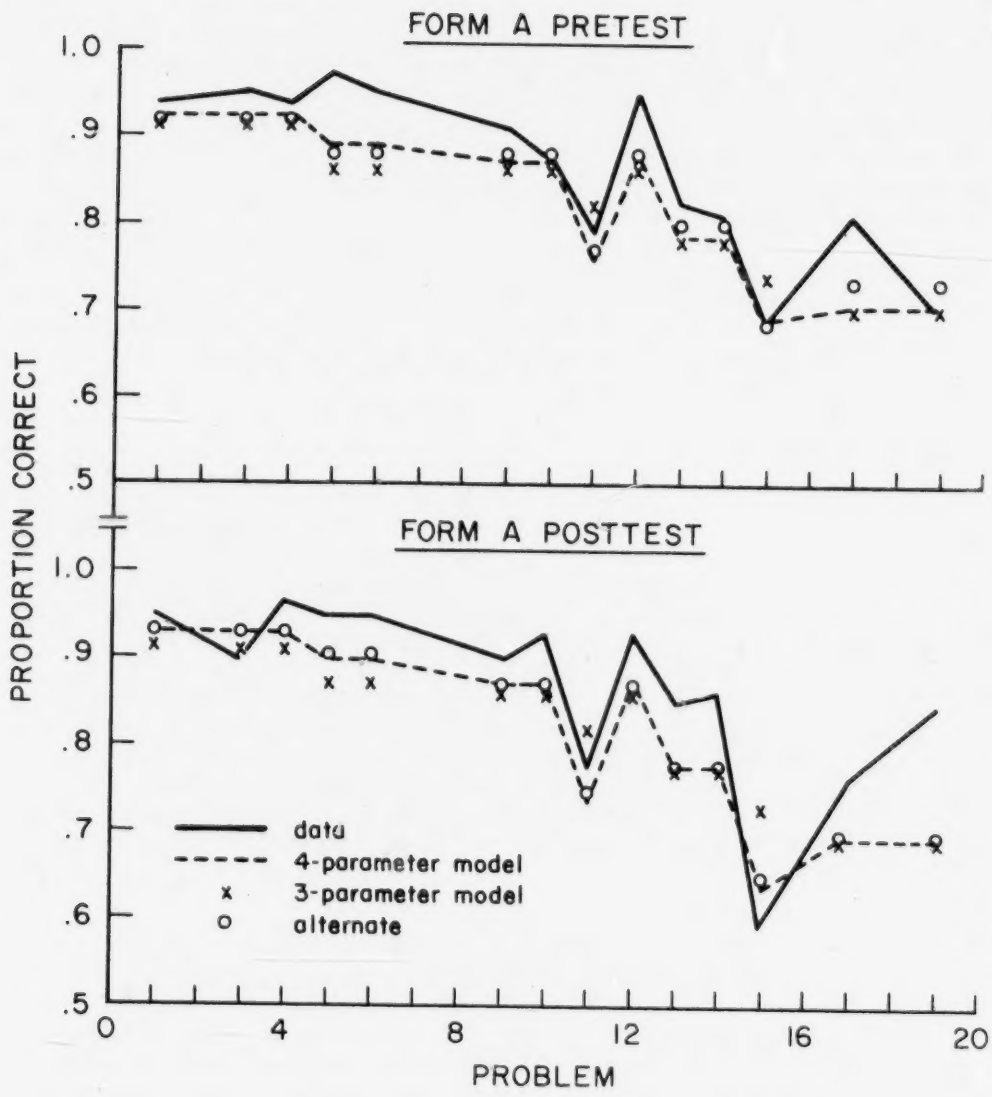

Fig. 1. Proportion correct per problem for data and models.

IV. CONCLUSIONS

Several conclusions emerge from this research over a period of several years on mathematical models of elementary-mathematics learning and performance. We break these conclusions into four broad classes.

1. Empirical Adequacy

The work indicated in this report, but reported more extensively in Suppes and Morningstar (1972), and in other publications that have not yet appeared, demonstrates that the kind of process models using ideas from the theory of automata and the abstract theory of register machines have direct application to the empirical study of students' performance in elementary arithmetic skills. It is fair to say that the models developed as the main focus of this project are probably the most detailed process models that yet exist in the psychological or educational literature, insofar as the basic skills of elementary mathematics are concerned. From the work performed it is fair to conclude that automaton models of the basic operations of addition, subtraction, multiplication and division can provide a detailed account of student performance, including especially the analysis of the typical errors that students make.

2. Open Questions

At the same time, the research begun under this project leaves more questions open than solved. The automaton models, which were most thoroughly tested empirically, do not provide anything like a satisfactory account of the perceptual processing engaged in by students, nor do they take adequate account of the actual real-time operations. The models are defined in terms of discrete time rather than continuous time, and no perceptual apparatus is assumed. Extensions in both of these directions are needed in order to develop more adequate models. The first steps of incorporating some simple perceptual processing were developed as part of the register machine models described in Section II, but these models represent only a very elementary and simple form of perceptual processing. The actual perceptual processing engaged in by students is obviously more elaborate and more complex in nature.

3. Need for Learning Models

Although it was the original intention of the project to develop a wide range of learning models as well as performance models, the details of the performance models developed and tested in this project were sufficiently complex and the problems of collecting adequate bodies of data were sufficiently difficult, that the next steps to correspondingly detailed learning models was not taken. Some work on learning was described in Suppes (1973), and further work on learning will be reported in some of the unpublished work yet to appear, but it is fair to say that the models as described above concentrate almost entirely on performance. The beginning of a complex theory of learning for finite automata in Suppes (1969) and in Rottmayer (1970) form a promising beginning but further work is needed for extension.

4. Practical Applications

In spite of the shortcomings of the models developed as described above, the automaton and register machine models developed already have potential applications in the organization of individualized curriculum in elementary mathematics, especially in a computer-assisted instruction setting. Perhaps the central criticism that can be made of most work in computer-assisted instruction at the present time is the relative simplicity of the model of the student assumed. The kind of models developed here, although still far too simple, do take a definite step toward postulating more than a simple error rate and a simple latency of response in the student, and do attempt to impose an internal structure on his processing of curriculum materials. As more deeply individualized curriculum is attempted, the kind of models developed as basic research under this project should play a natural part in providing tools for the development of such highly individualized curriculum.

The potential of the approach is that presenting exercises to the student on an individualized basis will no longer be based simply on the error rates and latency rates of the student, but on the estimate of detailed parameters of a model of his internal processing capacity. Errors will now be translated into error parameters that have a meaningful interpretation in a process model of the basic skills the student is learning. Applications of such detailed models would seem difficult in classroom settings, but it is considerably more practical when placed in a CAI setting, and there is reason to hope that such models will be developed during the coming decade.

V. BIBLIOGRAPHY

Buckingham, B. R. Adding up or down, A discussion. Journal of Educational Research, 1925, 12, 251-261.

Cronbach, L. J., & Suppes, P. (Eds.) Research for tomorrow's schools. New York: Macmillan, 1969.

Rottmayer, W. A. A formal theory of perception. Technical Report No. 161, 1970, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Shepherdson, J. C., & Sturgis, H. E. The computability of partial recursive functions. Journal of the Association of Computing Machinery, 1963, 10, 217-255.

Suppes, P. Stimulus-response of finite automata. Journal of Mathematical Psychology, 1969, 6, 327-355.

Suppes, P. Facts and fantasies of education. In M. C. Wittrock (Ed.), Changing schools: Alternatives from educational research. Englewood Cliffs, N. J.: Prentice-Hall, 1973.

Suppes, P., Hyman, L., & Jerman, M. Linear structural models for response and latency performance in arithmetic on computer-controlled terminals. In J. P. Hill (Ed.), Minnesota symposia on child psychology. Minneapolis: Univ. of Minnesota Press, 1967. Pp. 160-200.

Suppes, P., Jerman, M., & Brian, D. Computer-assisted instruction: The 1965-66 Stanford arithmetic program. New York: Academic Press, 1968.

Suppes, P., & Morningstar, M. Technological innovations: Computer-assisted instruction and compensatory education. In F. Korten, S. Cook & J. Lacey (Eds.), Psychology and the problems of society. Washington, D. C.: American Psychological Association, Inc., 1970. Pp. 221-236.

Suppes, P., & Morningstar, M. Computer-assisted instruction at Stanford, 1966-68: Data, models, and evaluation of the arithmetic programs. New York: Academic Press, 1972.

Thorndike, E. L. Educational psychology. Vols. 1, 2, 3. New York: Teachers College, Columbia University, 1913, 1914.

Thorndike, E. L. The psychology of arithmetic. New York: Macmillan, 1922.